Approved for public release; distribution is unlimited

# Data Fusion and Tracking System Testbed

Sept 23, 2000

Brian T. Mays
Hao Q. Vu
U.S. Army Research Laboratory
Adelphi, MD 20783-1197

## ABSTRACT

The Acoustic Signal Processing Branch of the U.S. Army Research Laboratory (ARL) is carrying on research into Battlefield target localization and tracking. To support the development and evaluation of new hardware and algorithms, ARL developed the Data Fusion Testbed and Tracking System (DFTTS). The DFTTS is a research and evaluation tool that enables researchers involved in developing sensors, localization, target detection, identification, and tracking algorithms to quickly evaluate their performance in a real-time environment. The DFTTS provides the necessary software and hardware backbone to support research and development efforts. Its major components are Sensor Signal Processing Nodes (SSPN) that can simultaneously host multiple sensor technologies and algorithms, along with a second-level Data Fusion Gateway Node (DFGN), which fuses SSPN data and data from other high-level sensor packages. This backbone allows new efforts to focus solely on algorithm and/or hardware development and minimize integration issues and time to in-field demonstration. Its architecture design is based on a distributed-multiprocessor, multitasked system. The design emphasizes a network of independent hardware and software processing modules, interconnected by loosely coupled communication links. This allows for such capabilities as real time comparisons of multiple detection and tracking algorithms and future upgrades of communication links. Since the DFTTS is a research and development tool, it includes methods for adjusting and tuning the system during its operational state and for monitoring of sensor data in real-time. This allows for on-the-fly experimentation with new software algorithms and hardware devices without the need to rebuild the system from the ground up. It also enables diagnostics to be performed on the system, so that the developer can make intelligent decisions about the functional validity of the software and hardware being tested and developed.

# Form SF298 Citation Data

| Report Date ("DD MON YYYY") 23092000 | Report Type N/A | Dates Covered (from... to) ("DD MON YYYY") |
|---|---|---|

| | |
|---|---|
| **Title and Subtitle**<br>Data Fusion and Tracking System Testbed | **Contract or Grant Number** |
| | **Program Element Number** |
| **Authors**<br>Mays, Brian T.; Vu, Hao Q. | **Project Number** |
| | **Task Number** |
| | **Work Unit Number** |
| **Performing Organization Name(s) and Address(es)**<br>U.S. Army Research Laboratory Adelphi, MD 20783 | **Performing Organization Number(s)** |
| **Sponsoring/Monitoring Agency Name(s) and Address(es)** | **Monitoring Agency Acronym** |
| | **Monitoring Agency Report Number(s)** |
| **Distribution/Availability Statement**<br>Approved for public release, distribution unlimited | |
| **Supplementary Notes** | |
| **Abstract** | |
| **Subject Terms** | |
| **Document Classification**<br>unclassified | **Classification of SF298**<br>unclassified |
| **Classification of Abstract**<br>unclassified | **Limitation of Abstract**<br>unlimited |
| **Number of Pages**<br>14 | |

## 1.0 BACKGROUND

The Acoustic Signal Processing Branch of the U.S. Army Research Laboratory (ARL) is carrying out research into Battlefield target tracking with Unattended Ground Sensors (UGS). In support of these efforts, both sensor technologies and signal processing algorithms are evaluated for performance and utility so that new capabilities can be added to UGS. Once a sensor or algorithm has been tested in the laboratory, the next phase of engineering evaluation is field experiments with live targets and realistic scenarios. The step from the laboratory tests to field demonstration can be significant, since the new sensor or algorithm is typically a small part of the overall concept being demonstrated. For example, a researcher may want to add a magnetic detection capability to a sensor field to see if this improves the ability to discriminate and track multiple vehicles in a Battlefield scenario. The desire to quickly conduct such field evaluations motivated ARL to develop the Data Fusion Testbed and Tracking System (DFTTS).

## 2.0 OVERALL SYSTEM ARCHITECTURE

The DFTTS provides the system-level backbone required in a networked field of UGSs. The DFTTS is made up of three key components: the Sensor Signal Processing Node (SSPN), the Data Fusion Gateway Node (DFGN), and the Combat Information Processor (CIP).

Figure 2.1 shows the overall system configuration for the DFTTS. A group of remote sensor processing systems (SSPN and UGS) locate in a geographically related area are loosely tied together via low-bandwidth radio links to a central DFGN, where various tracking algorithms can be used to correlate and process the information to form target tracks. The DFGN sends its results to a ground station for display at a remote CIP. The link between the DFGN and the CIP will also be a low-bandwidth radio link.

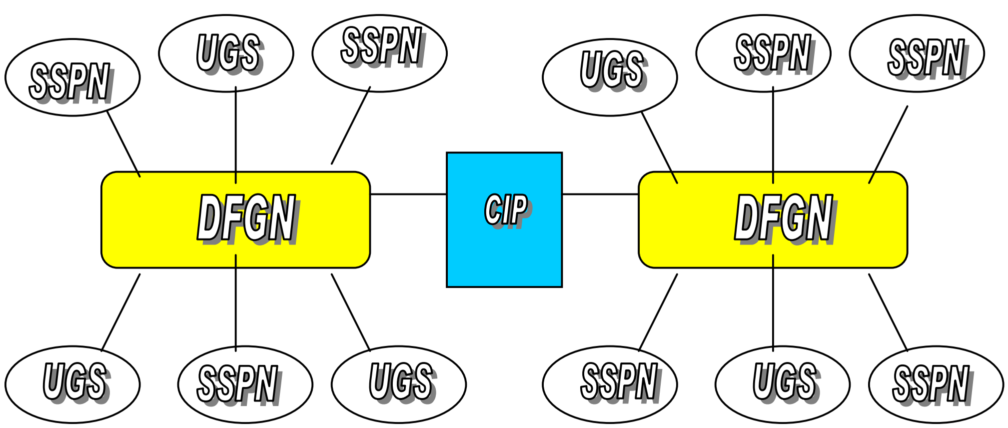


**Figure 2.1** Overall system architecture

## 2.1 Sensor Signal Processing Node Functions

The SSPN is an automated intelligent sensing system that contains a local processing unit to survey its local environment and attempt to detect known and unknown targets in its listening or viewing area. Generic information such as what is detected and at what time and location are computed locally and eventually forwarded to the DFGN. The goal of this system design is to provide for an expandable testbed where different types of sensor technologies (both current and future) can be deployed in a single SSPN with minimal disruption to the original system design. Along with simultaneously hosting multiple processing algorithms, the SSPN can serve as an algorithm server to remote MATLAB clients. The MATLAB clients allow researchers to evaluate new algorithms in the full-up system without porting to an embedded solution. This approach is especially useful in early development as it allows one to determine the merits of a specific algorithm before full integration into the system. In addition, as new types of network communications technologies are developed, it is desirable that the SSPN design to be adapted easily without the need to redesign the entire system.

## 2.2 Data Fusion Gateway Node Functions

Each DFGN will control up to 32 SSPN units in a geographical area covering several kilometers. The function of the DFGN is to accept target detection, location, and identification information from the different SSPN and UGS units associated with it and correlate the information for tracking purposes. The DFGN will also act as a communication router between its UGS units, which contain information similar to SSPN units but not requiring identical SSPN's emulation, and the CIP located at the ground control station where the actual man-machine interface will be implemented. Like the SSPN, the DFGN will be used as a testbed for testing various tracking algorithms and communication systems. The design of the software architecture must be sufficiently modular to allow for the inclusion of multiple tracking algorithms and communication devices without disruption to the overall system design. It is desirable to include and test new technologies (hardware and software) as these technologies become available in the future. Evaluation modules will be part of the DFGN to help assess the performance of each tracking algorithm. These evaluation modules could be implemented as part of the running real-time system, or they could be implemented as post- processing tasks that examine data logged by the system.

## 2.3 Combat Information Processor Functions

Serving as the man-machine interface to various SSPNs, UGSs, and DFGNs will be via an ARL developed combat information processor (CIP). The CIP will display relevant map data to the area being surveyed by the SSPN/UGS/DFGN system. It will allow a human controller to modify algorithm parameters in real time to adjust tracking performance. The CIP will control the analog data archiving system as well issuing status queries to the SSPNs and DFGNs. The CIP will provide tracking displays, raw data displays, terrain features, land elevation information, and other pertinent information to help a user make an intelligent assessment of the combat situation being observed.

## 2.4 Information Flow

The flow of information between processing modules takes two forms. There is 1) raw data such as sensor target detection information or track calculation data and there is 2) control/status information such as recording data control commands or algorithm parameter changes.

As a general rule, data flow from the sensors back to the DFGN and eventually to the CIP. Figure 2.2 shows the overall flow of raw data passing between the different software modules. Raw data are usually presented in the form of large blocks of binary information coming from an analog to digital (A/D) conversion unit. The raw data are processed by an algorithm task and possibly archived in its digital form for later testing of the algorithm task. The algorithm task will put out target detection information to pass to the DFGN. If appropriate, the data output task could also pass this information to another local process for graphic display the results for algorithm verification purposes.

When the target detection information reaches the sensor report data logger at the DFGN, the information will be optionally archived to a file for later testing and evaluation of the tracking algorithms. The sensor report data logger will pass the data to the tracking algorithm processors (and optionally to a raw sensor report data display task). The output of the tracking algorithms will go to the track data logger.

The track data logger will optionally archive its data input to a file for later analysis. Its primary responsibility is to act as a gateway into the CIP software architecture by reformatting the data as necessary and passing them to CIP for display. UNITCTRL will take care of displaying the track information on the CIP generated and controlled mapping system.

Control information generally originates from the CIP (usually from the user) and flows to the DFGN and then to the SSPN. Control information has the possibility of touching every processing module in the system. Minimally, there can be control requests to activate diagnostic modes during software development. Other possible controls include setting algorithmic parameters in the target detection and tracking modules, modifying display parameters, and activating/deactivating various data archives along the processing path. Requests can be generated to turn on and off various display features of the system.

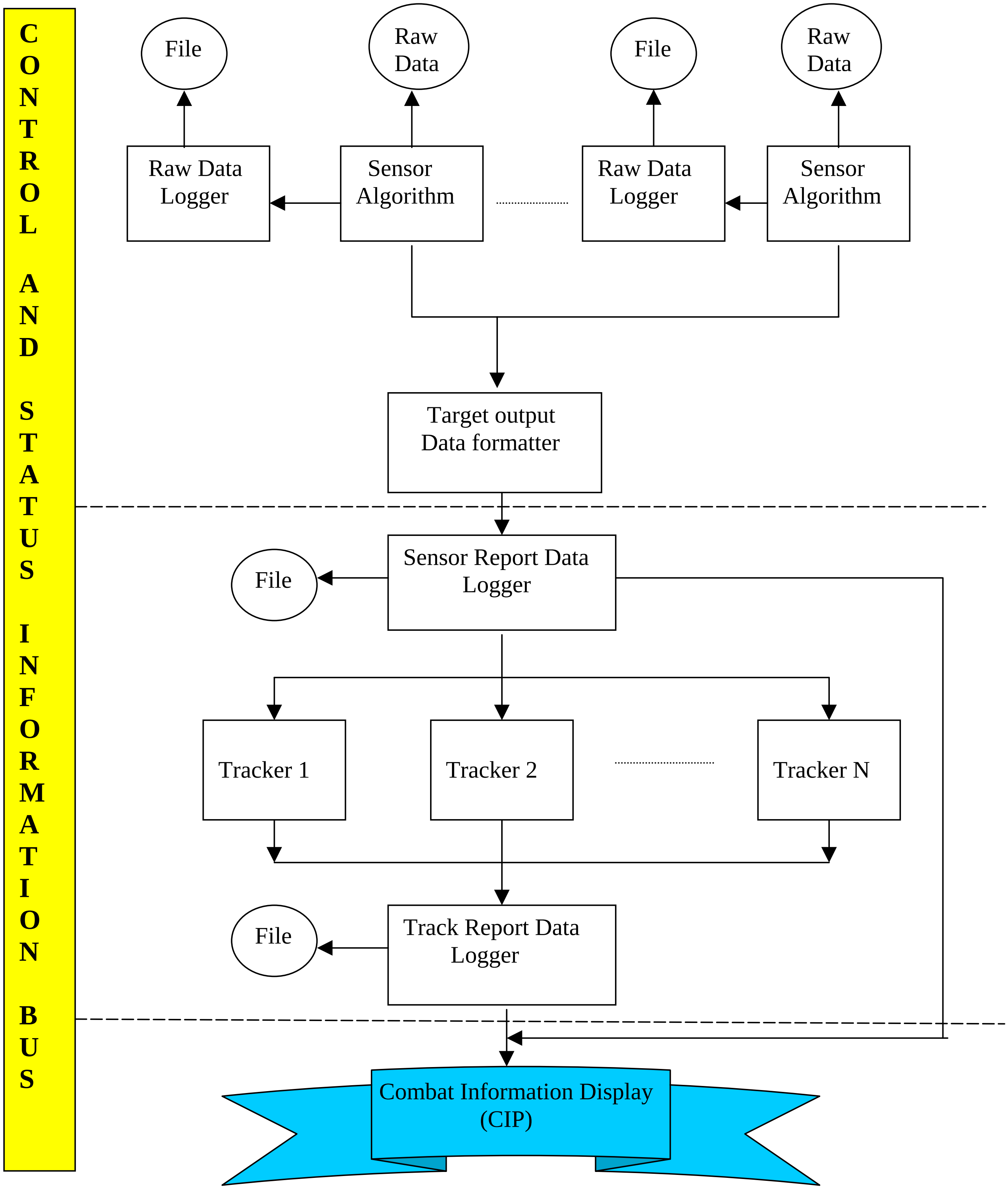


**Figure 2.2** System Information Flow

Status information flows back to the CIP usually triggered by a user control request, but unsolicited status information may also be sent. Status requests may be for records of data progress checks, algorithm performance progress reports, unexpected system failure reports and network connectivity status checks. Status information can come from any processing module and will be displayed and/or logged back at the CIP.

# 3.0 SENSOR SIGNAL PROCESSING

## 3.1 Overall Architecture

Figure 3.1 shows the SSPN overall system configuration.

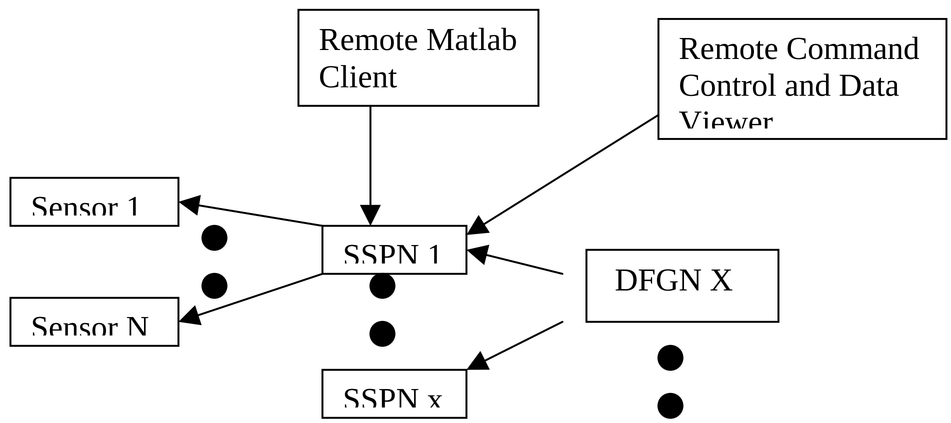


**Figure 3.1** SSPN Overall System Architecture

## 3.2 SSPN Hardware

The SSPN was designed with several key requirements established from the outset. The Testbed must support a large number (56) of analog input channels, host local and remote processing algorithms, provide data recording capabilities, work reliably in adverse environmental conditions, and run from a single DC power source. All mechanical media must be removable and the units must be self-contained for shipping purposes and to ease setup on site.

The SSPN consists of two types of sub-assemblies: the Main Control Unit (MCU) and Analog Signal Conditioning Boxes (ASCB). The complete system will consist of one main unit and up to eight ASCBs. Each ASCB groups the channels in two blocks of four for the purpose of programmable gain and cut-off frequency control. The ASCBs are designed to be fully controlled by the MCU in the system using a simple four-wire interface to each box. This command and control capability is a key feature in removing human error from the setup phase of operation. This programmability also allows for automatic gain control or adaptive bandwidth control if so desired. The MCU is a self-

contained assembly with several internal components. The main unit provides a climate-controlled environment for the internal components and external connections to the ASCB, main system power, and diagnostic port for the unit. The internal components consist of a Compact Peripheral Computer Interface (CPCI) Cage, Packet Radio, wireless LAN radio, Removable SCSI Drive, PLGR/GPS Receiver, Climate Control Circuit and DC/DC power Converter Unit. The system layout appears in the Figure 3.2.

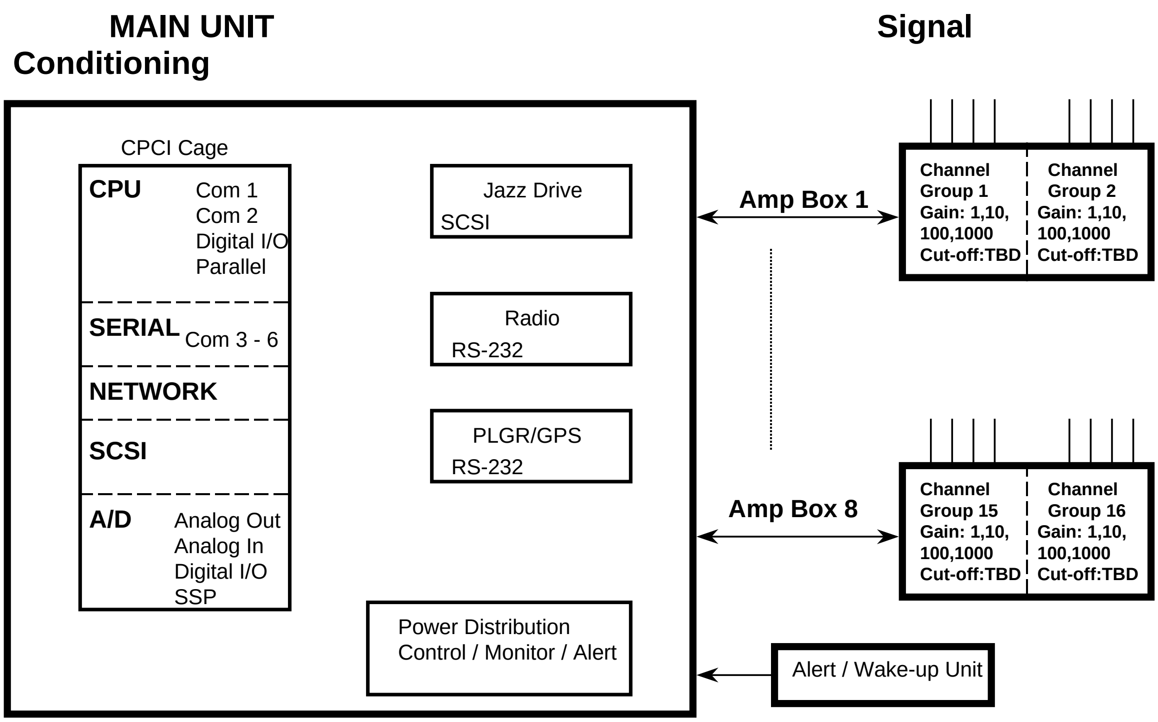


**Figure 3.2** SSPN Hardware System Architecture

In addition to the local SSPN hardware, remote assets may also be attached to the system via the wireless LAN connection. These remote assets are typically command and control and remote processing computers. More detail is provided section 3.3.

## 3.3 SSPN Software Architecture

The SSPN software architecture design is based on a distributed multiprocessor, multi-tasked system. The design emphasizes a network of independent software processing modules to be inter-connected by an inter-process communication link. With this architecture, different applications can be added or removed without adversely effects the other areas of the Testbed. The design philosophy of the software architecture is that the system be modular, with independent software tasks and well-defined inputs and outputs. The software is designed to allow new algorithms to be quickly implemented as stand-alone processes. These additional algorithms simply attach to several SSPN resources, which provide full integration into the SSPN system.

The SSPN can support 56 channels of analog data. These channels are organized into groups of 8, clustered into individual ASCBs. The data groups can have different sample rates to allow multi-rate processing and minimize data throughput requirements. A group allocation file limits which algorithms can request data from specific groups to ensure algorithm isolation. The algorithms can be allocated multiple groups if more than 8 channels of analog data are required. The current SSPN can support up to 10 independent algorithms. The actual number of concurrent algorithms may be lower due to processor loading.

## 3.4 Sensor Processing System Architecture

The SSPN resources available to the algorithms include a data-server, an inbound message dispatcher, an outbound message system and global SSPN parameters. Figure 3.4 shows the main processes running on the SSPN and their basic relationships. The main processes are the Data Server, Communication Manager, Dispatcher, Algorithm Server, Remote Monitor, and Recorder.

The Data Server manages the Analog to Digital converter asset and provides analog sensor data to the processing algorithms, while maintaining separation between the algorithms. The Data Server will process data requests and requests to change gains and cutoff values for the Analog Groups. The algorithm-based data requests are asynchronous to the data acquisition process so requesting algorithms receive the most recent data available. This approach allows algorithms that cannot keep up in real time to block process the data and still produce periodic answers. For example, an algorithm may request a block of data and take 5 seconds to process the information. The algorithms next request for data will be the most recent block of data acquired that is not 4 seconds old. When algorithms are keeping up in real time, requests made before the receipt of a new block of data are held until the data request can be fulfilled.

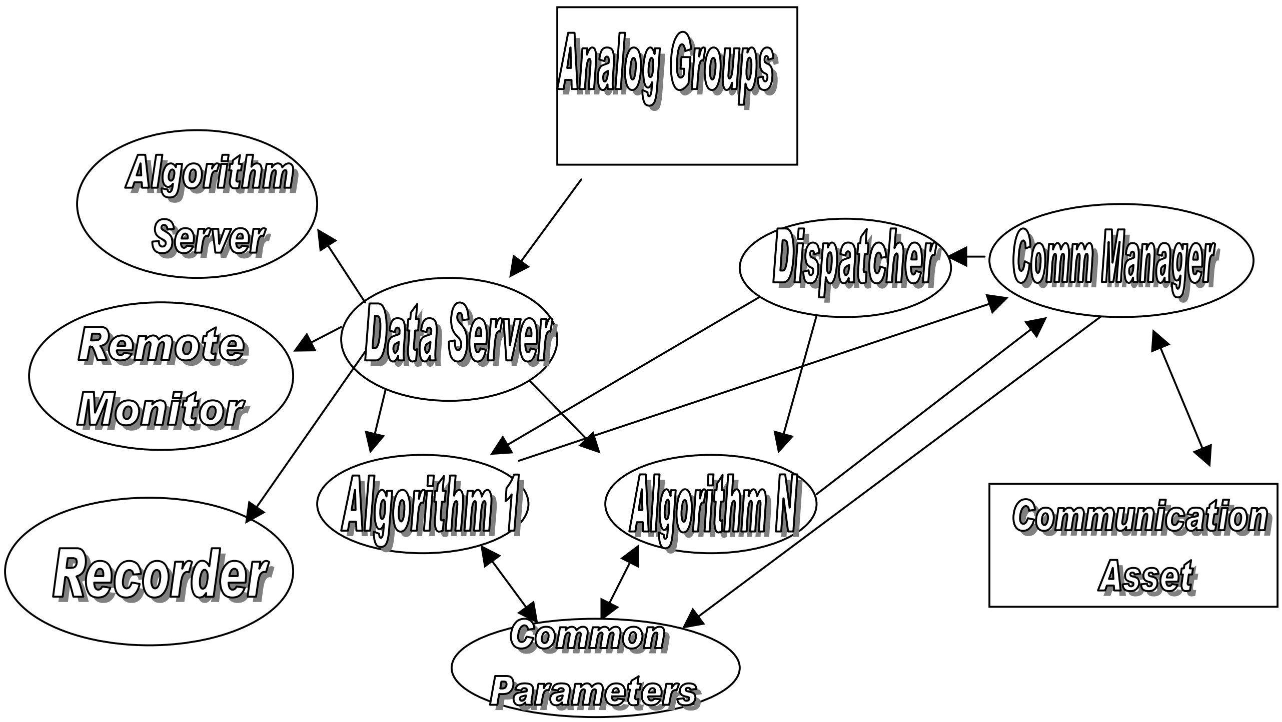


**Figure 3.4** SSPN Software System Architecture

The data available to algorithms come in two forms: raw and Fast Fourier Transform (FFT) data. Algorithms request data by groups from 1 to 8 with each group containing 8 channels of analog data. When multiple groups are requested by a single algorithm, the data server attempts to provide data from the same time window across the groups. Since the data acquisition and data requests are asynchronous processes, it is possible that all of the blocks are not from the same time window. In the rare case that this occurs, the requesting algorithm should re-request the multiple groups to assure the times coincide. The data is passed to the requesting process with each group in a separate array. Each array of data contains a header followed by a data block.

The Communication Manager is the process that maintains an external communications link to the DFGN. The manager has a simple message queue interface with the algorithms that allow arbitrary messages to be passed to the DFGN. The only requirement on the message format is that the top nibble of the first byte represents the task id of the process sending the message; this allows proper interpretation by the DFGN. The final role of the Communications Manager is to provide three different physical link options to the DFGN, which are transparent to the hosted algorithms on the SSPN.

The Dispatcher process is closely tied to the communications manager and routes inbound messages to the appropriate algorithms. The inbound messages again can be arbitrary as long as the top nibble of the first byte represents the task id of the destination

algorithms. Algorithms need to attach to the dispatcher and provide a child parser process to handle routed messages.

The Algorithm Server allows remote clients to attach to the SSPN system and operate as if they were integrated into the SSPN via the wireless TCP/IP connection. The Algorithm Server has facilities that allow the remote clients to request data from the SSPN, control the ASCBs attached to the SSPN, and send reports back to the DFGN via the SSPN radio. This allows the results of the algorithms to propagate to the rest of the system as if the algorithm were hosted locally on the SSPN. The Algorithm server allows any TCP/IP capable client to attach to the SSPN. The client is intended to allow higher level languages such as MATLAB and LABVIEW to be used as a processing engine in the SSPN system. The remoting of the client also provides two additional benefits to system performance. First it allows much deeper visibility into the remote algorithm since the remote application will have all of its display capabilities available on its hosting platform. Second the remote algorithm can be integrated without reducing overall system reliability, because no additional processor loading occurs and software crashes are isolated to the client. A final purpose of the remote client is to allow state of the art computer power to be applied to a problem if required without requiring the base SSPN to be upgraded. As mentioned above, one key desire was to host native MATLAB m-files in the system. At the time of development, MATLAB did not provide native TCP/IP support. To solve this problem ARL also developed a library of MEX files along with m-file rappers to allow direct access to the SSPN's Algorithm server from within standard m-files.

The Remote monitor is also TCP/IP based and operates over the wireless TCP/IP connection. It has a companion client, which is used for data monitoring during the SSPN operations. This task is similar to that performed by the algorithm server except that data can only be viewed and no control of the ASCBs is possible. This limitation ensures that the data are not changed. The Remote monitor allows complete data visibility into the SSPN for data integrity monitoring or real-time sensor analysis. The monitor client was developed under LABVIEW to provide real-time data plot of specific analog channels or spectral waterfall plots of a specific analog channel.

The Recorder task supports various commands that allow the remote controlling application to record specific data and monitor items such as disk space, channels being recorded, fully reconfigure the unit for changing system parameters such as sample rate etc. ARL developed a LABVIEW command and control application to operate as a main system console for multiple SSPN's. The recorder can be used in stand-alone data collection exercises or for background data achieving during full up field experiments.

The final asset available to the hosted algorithms is a common parameter table in the form of shared memory. These common parameters are updated directly by the communications manager and contain system parameters that all algorithms should adhere to. Examples of these common parameters are report rates, report enable/disable status, etc.

## 4.0 CONFIGURATION AND CO-EXISTANCE ISSUES

While the design of the SSPN allows the hosted algorithms to operate independently, the system configuration cannot be solely based on the unique requirements of each algorithm. The need for compromise arises in the selection of sample rates for each algorithm. The SSPN has a single analog to digital converter asset and must be configured so as to allow proper operation.

The SSPN design allows multi-rate data acquisition across groups of data, which is transparent to the hosted algorithms. To achieve this, the system must ensure the sample rates between the groups are integer related. A second consideration in selecting the sample rates is whether any algorithm will be requesting pre-processed FFT data. If this is the case, the sample rates must also be a power of 2. The system selects the different samples rates between groups according to a group divisor specified for each group in the system. The group divisor can be a value between 1 and 128. If the above conditions are not met, the system will attempt to run but may eventually fail due to buffering problems.

In configuring the group sample rates, first a master sample rate must be selected, then all the group divisors can be selected to match individual algorithms needs. Again, if FFT data are to be provided the sample rate must be a power of 2. The next option, the update rate, determines the data block sizes passed around the system. The value represents how many blocks per second will be used. This parameter also affects the FFT data and represents the bin width in Hertz. For example, given an update rate of 4, the data blocks span 250 mSec and the FFT resolution is 4 Hz.

The final configuration option is a group allocation table. This table is used to control which groups an algorithm can request and manipulate through the data server. This table serves as a protection device to keep rogue algorithms from affecting others in the system. Multiple algorithms can have common access to a common group of data but they must be aware of the other's actions. In this case the algorithms are slightly integrated in the sense that they are aware of each other.

## 5.0 DATA FUSION GATEWAY NODE

### 5.1 DFGN Hardware

The DFGN is an SSPN with a communication hub attached to it. Therefore, the DFGN hardware is exactly the same as that of an SSPN. Because of this flexible hardware configuration, any SSPN can be designated as the DFGN. The DFGN is designed to handle dual purposes: to serve as an SSPN and to serve as a data fusion center with a routing capability. The DFGN is often referred to as the Sensor Signal Processing "Mother" node. However, because of the low processing power in the current SSPN, the DFGN is limited to its data fusing and routing capabilities.

The communication hub contains a series of low bandwidth packet radios that are used to communicate with the SSPNs and UGSs. The radios are linked to the CPU via a terminal server (which used to convert RS-232 protocols to the TCP/IP protocol) and an Ethernet hub. The DFGN system layout is showed in Figure 5.1.

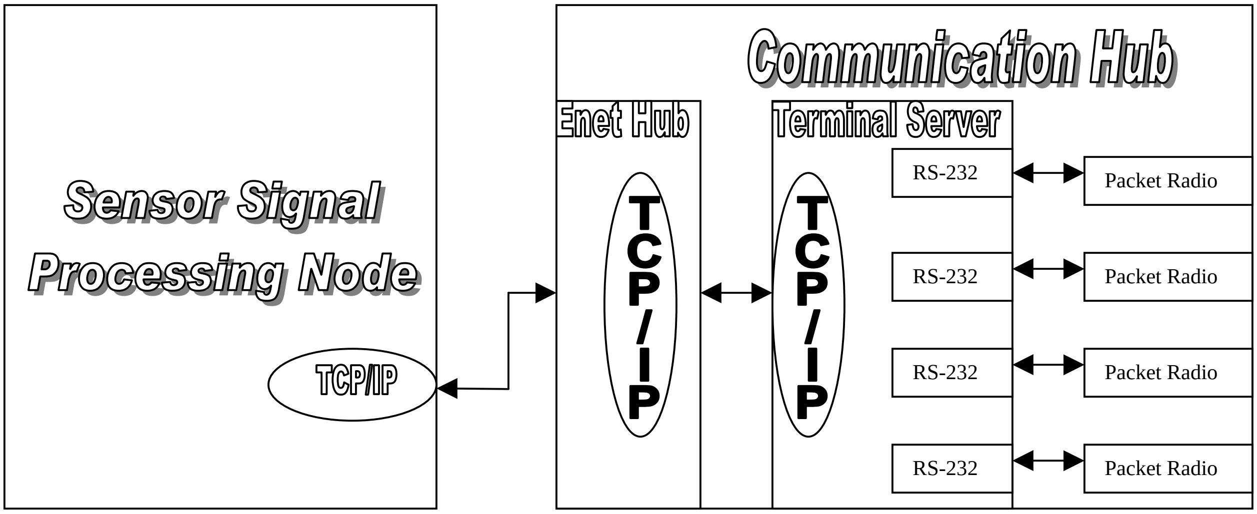


**Figure 5.1** DFGN Hardware architecture layout

## 5.2 DFGN Software

Figure 5.2 shows the information flow between the various software modules in the DFGN. Radio drivers (and actually any hardware communication device driver) will map device dependent characteristics to the communications network to a device independent form for the radio network router. All messages to and from the outside will pass through the radio network router.

There will be at least one tracker process (and possibly several) in the system. Each tracker will be able to handle a specific set of data formats. These data formats will be generated by the SSPNs and UGSs attached to the overall system and will be known also by the sensor report data logger. A tracker must attach to the sensor report data logger and indicate to the logger the type of data formats that it is interested in seeing. When data arrive from the communication network, they are routed to the sensor report data logger. From here, the raw data will be routed to all trackers that have requested the data type. Each sensor report will be tagged with a unique identification number so that post- processing routines can be used to evaluate the tracking algorithm performance.

The trackers will perform data specific tracking algorithms and the system will eventually produce a report of what it concludes is out in the area being surveyed by the SSPNs. These results will be sent to the CIP process, UNITCTRL, for graphic display. Tracker output will be passed to the track data logger in a generic format. This output will include unit identification, track identification, track creation, track deletion and extension commands.

Confidence levels concerning unit classification and unit location will also be included in the output. Each time a tracker generates an output concerning unit location and/or identification, it will also include a list of the sensor reports that it used to make this deduction. Post processing routines will use this information along with a ground truth file to evaluate the tracker performance. Tracker output that is not necessary for real-time display will not be sent over the radio network to the CIP. This information will be logged to a data file local to the gateway for later evaluation.

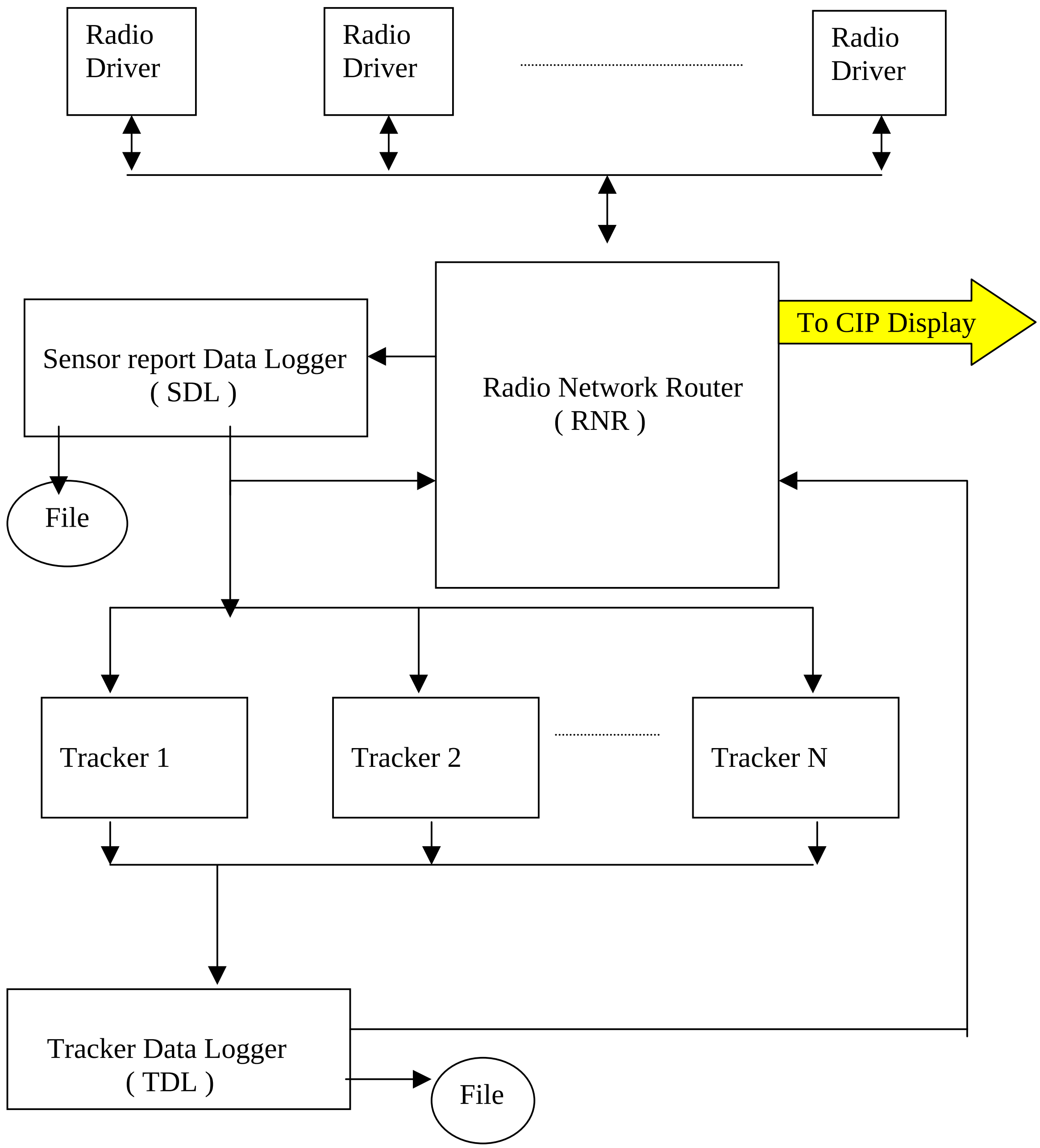


**Figure 5.2** GWDFP Software Processes Flowchart

## 5.3 Inter Process Communication (IPC)

In order to maintain its desire for expendability and modularity, the DFGN choose TCP/IP as its inter-process communication link among its tasks. For ease of use and reducing the development time in implementing new tracker algorithms into the DFGN, Army Research Laboratory (ARL) has provided the developers an interface library that operates over a TCP/IP network. This interface library will hide all the communication details from the developers. Therefore, it will allow the developers to attach their algorithms to the DFGN painlessly and quickly.

Note: All the library and network functions are developed in ANSI C.

# 6.0 CONCLUSIONS

In the past, transitioning new sensors and algorithmic concepts from the early stages of simulation to technology demonstrations has been costly and time consuming. In many cases, the demonstration phase may be cost prohibitive for small or incremental system improvements, even when the new developments show great promise in increasing system performance. These past limitations motivated ARL to develop the DFTTS in support of its ongoing research into Battlefield target tracking systems. The DFTTS provides the backbone required to host new sensors and algorithms at their earliest stages of development. The use of the DFTTS provides an efficient mechanism to add new technologies and approaches quickly and to determine their direct impact at the complete system level in real-time. This capability greatly reduces the design cycle time between laboratory simulation and field experiments for user evaluation. The DFFTS has proven its utility in recent field experiments by allowing the rapid integration of third party sensors, MATLAB based algorithms, C-coded algorithms, and independent sensing systems augmenting the processing at even higher levels.